L Number	Hits	Search Text	DB	Time stamp
1	1	10/700621	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/08 11:17
-	1132	((714/30,34,39,45) or (717/128)).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:20
-	0	"instruction adj address" same start same stop same trace	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 15:53
-	0	"instruction adj address"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:23
-	0	"instruction adj address" same start same stop same trace	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 15:52
-	3	"instruction address" same start same stop same trace	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 15:54
-	7140	"instruction address"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:23
-	5059	(begin$4 or initiat$4 or start$4) same (end$4 or complet$4 or stop$4) same trace	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:21
-	452	relative adj branch	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/22 11:28
-	24	((begin$4 or initiat$4 or start$4) same (end$4 or complet$4 or stop$4) same trace) and (relative adj branch)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 16:05
-	11024	execution adj time	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 16:05
-	171	((begin$4 or initiat$4 or start$4) same (end$4 or complet$4 or stop$4) same trace) and (execution adj time)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/22 11:28
-	112	register and (((begin$4 or initiat$4 or start$4) same (end$4 or complet$4 or stop$4) same trace) and (execution adj time))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 16:07

-	53	debug$4 and (register and (((begin$4 or initiat$4 or start$4) same (end$4 or complet$4 or stop$4) same trace) and (execution adj time)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/07 16:08
-	0	08992610.an.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/13 14:47
-	2	("6154857").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/11/13 14:47
-	1311	((714/30,34,39,45) or (717/128)).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/22 11:38
-	36	relative adj branch and @pd>20021114	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/22 11:34
-	175	(((714/30,34,39,45) or (717/128)).CCLS.) and @pd>20021114	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/22 11:38
-	1623	((714/30,34,39,45) or (717/128)).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:21
-	2739	((714/30,34,39,45,38) or (717/128)).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:21
-	7420	(begin$4 or initiat$4 or start$4 or enter$) same (end$4 or complet$4 or stop$4 or exit$4) same trace	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:36
-	0	"instruction adj address"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:23
-	9946	instruction adj address	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:23
-	65	((begin$4 or initiat$4 or start$4 or enter$) same (end$4 or complet$4 or stop$4 or exit$4) same trace) same (instruction adj address)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:23
-	63398	((begin$4 or initiat$4 or start$4 or enter$) same (end$4 or complet$4 or stop$4 or exit$4)) same address	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:37

-	532	(((714/30,34,39,45,38) or (717/128)).CCLS.) and (((begin$4 or initiat$4 or start$4 or enter$) same (end$4 or complet$4 or stop$4 or exit$4)) same address)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:55
-	1	("20040153802").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/07 19:57
-	2	("20020091494").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/08 11:17

Web Results **1** - **10** of about **118,000** for **real-time trace start address end**. (**0.87** seconds)

Contents
... Perform a Manual Tape Switch; ZTRAC-**Start** Macro Tracing for the **Real-Time Trace** Utility; ZTRCE-**Trace** Selected Records; ZTRMT-Remount Tape; ZTRTE-Manage IP ...
publib.boulder.ibm.com/infocenter/tpfhelp/topic/com.ibm.tpf.doc.4.1-PUT18/gtpo1/gtpo1m02.htm - 76k - Cached - Similar pages

ZCNTM-**Start** Macro Counting for the **Real-Time Trace** Utility
... associated with specified file addresses, where fileaddr is a 4- or 8-byte file **address**. ... See ZTRAC-**Start** Macro Tracing for the **Real-Time Trace** Utility for ...
publib.boulder.ibm.com/infocenter/tpfhelp/topic/com.ibm.tpf.doc.4.1-PUT18/gtpo1/gtpo1m2p.htm - 15k - Cached - Similar pages
[More results from publib.boulder.ibm.com]

Metalink, MetalCE, High **End**, High Performance, 8051, In Circuit ...
... Display contents captured in **real time** and displayed in ... **Trace** Trigger Positions - **Start**, Center, **End**, Variable. **Trace** Collection Control - **Trace** ON, **Trace** OFF: ...
www.testech-elect.com/metalink/mi_ta_500.htm - 26k - Cached - Similar pages

Tracedmp.exe: **Trace** Dump
... TraceDmp can also poll **real-time trace**-buffer data and ... The procedure for doing **real-time** tracing is similar to ... double-click the file to **start** the installation ...
www.microsoft.com/windows2000/techinfo/reskit/tools/existing/**trace**dmp-o.asp - 35k - Cached - Similar pages

Logman
... rt Turns off the **real-time** logging option ... active buffers of an existing event **trace** session to ... To **start** collecting data for collection queries, log Performance ...
www.microsoft.com/windowsxp/home/using/productdoc/en/NT_Command_Logman.asp - 29k - Cached - Similar pages
[More results from www.microsoft.com]

I2C Bus Monitor
... **Real-Time Trace** to 100Kbit/s. Supports General Calls, and 7 bit ... **End** of Buffer, Display -**START** OF **TRACE**- MMMM:**START** AA RA ... AA DD C R/W N/A, **Trace** Message Number ...
www.mcc-us.com/101.htm - 11k - Cached - Similar pages

MetaLink 8051 MetalCE-XF-DS80C400 Specifications
... Additional **Trace** Display contents captured in **real time** and displayed in an 8 frame cycle: ... **Trace** Trigger Positions - **Start**, Center, **End**, Variable. ...

www.metaice.com/8051-MIXF-DS400/DS80C400_Specs.htm - 22k -
Cached - Similar pages

MetaICE-XF-DS80C400
... System Access). Additional **Trace** Display contents captured in **real time** and displayed in ... 32-bit Time Stamp. **Trace** Trigger Positions - **Start**, Center, **End** ...
www.abtronix.com/metaice%20XF%20DS.htm - 16k - Cached - Similar pages

XP Logman commandline utility
... m <[**start**] [stop]> Change to manual **start** or stop rather than a scheduled begin or **end** time. ... [-]rt Run the Event **Trace** Session in **real-time** mode. ...
www.windowsnetworking.com/kbase/WindowsTips/WindowsXP/UserTips/Utilities/XPLogmancommandlineutility.html - 31k -
Cached - Similar pages

Re: [Gcl-devel] Re: (random tester) Another compiler side effects ...
... My guess > > is that a **trace** of compiler::add-info ... Loading gazonk1.o > **start address** -T 0x89c62a0 Finished loading gazonk1.o > **real time** : 0.190 secs ...
lists.gnu.org/archive/html/ gcl-devel/2003-11/msg00011.html - 11k -
Cached - Similar pages

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

real-time trace start address end Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Web Results **11** - **20** of about **118,000** for **real-time trace start address end** . (**1.66** seconds)

[PDF] DProbeST10 DBoxST10
File Format: PDF/Adobe Acrobat - View as HTML
... DBoxST10 Will Cover All Your Needs **Real-time trace** with 48 ... controlled **start**/stop of non statistical **realtime** code coverage ... 1 DBox-2 DBox-3 **Trace** memory Depth ...
www.hitex.com/pdf/DProbeST10.pdf - Similar pages

noral - press releases
... a pseudo sync point at the **start** of the ... This means that the complete **trace** buffer is always ... You can then keep executing your target in **real time** 'chunks' until ...
www.noral.com/app3.htm - 15k - Cached - Similar pages

Debugging with GDB: Tracepoints
... program's correctness depends on its **real-time** behavior, delays ... gdb) **trace** *my_function // EXACT **start address** of function ... bar (gdb) pass 2 (gdb) **trace** baz (gdb ...
davis.lbl.gov/Manuals/GDB/gdb_10.html - 41k - Cached - Similar pages

Re: [Gcl-devel] Re: (random tester) Another compiler side effects ...
... My guess is that a **trace** of compiler::add-info would be ... Loading gazonk1.o **start address** -T 0x89c62a0 Finished loading gazonk1.o **real time** : 0.190 secs ...
lists.gnu.org/archive/html/ gcl-devel/2003-11/msg00002.html - 9k - Cached - Similar pages
[More results from lists.gnu.org]

Metalink IM-8051-RA **Real Time** In Circuit emulator for 8051, Dallas ...
... **Real-time trace** with "view while executing code" **Trace** Buffer Size 16 ... Bus and 8 bits External Clips Variable **Trace** Trigger settings **Start**, Center, **End** ...
www.testech-elect.com/metalink/im_ra.htm - 17k - Cached - Similar pages
[More results from www.testech-elect.com]

The **Trace** Server on JNOS
... to do is activate it by using the **start** command above ... you want to "follow" the output of the file in **real time**. ... one command to instruct the server to **trace** on a ...
ka1fsb.home.att.net/**trace**srv.html - 23k - Cached - Similar pages

[PDF] UniSTAC-II SH4-JTAG Product Brief
File Format: PDF/Adobe Acrobat - View as HTML
... **start** and **end address**) Data **trace** within a specific range (specify **start/end address**, memory access ... supports the following compilers and **real-time** OS: Renesas ...
www.directinsight.co.uk/downloads/ eso/222/SH-Mobile-US2J_brief.pdf - Similar pages

System Administration: The Linux **Trace** Toolkit

... present data about the whole system: **trace start** time, **trace** ... de Montréal, where the Linux **Trace** Toolkit is ... courses on Linux internals and **real-time** derivatives ...
www.linuxjournal.com/article.php?sid=3829 - Similar pages

OMEGAMON II for CICS
... CICS does everything you expect from the leading **real-time** and historical ... **Trace** records include ... Task **start**; Entry and completion of EXEC CICS commands; Entry and ...
www.candle.com/www1/cnd/portal/CNDportal_Channel_Master/0,2938,2683_3902229,00.html - 38k - Cached - Similar pages

ND-78K4 EXPLANNING
... function, **Start**, Same as the section **trace start** event. Stop, Same as the section **trace** stop event. ... Internal RAM **real-time** display(Only Internal high-speed RAM ...
www.ndk-m.co.jp/asmis/eng/78k4h.html - 8k - Cached - Similar pages

Gooooooooooogle
Result Page: **Previous** 1 2 3 4 5 6 7 8 9 10 11 **Next**

real-time trace start address end Search

Search within results | Language Tools | Search Tips

Google Home - Advertising Programs - Business Solutions - About Google

Your search matched **4** of **1058483** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

trace <and> (start <or> entry) <and> (point <or> addr Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Action refinement for true concurrent real time**
Majster-Cederbaum, M.; Jinzhao Wu;
Engineering of Complex Computer Systems, 2001. Proceedings. Seventh IEEE International Conference on , 11-13 June 2001
Pages:58 - 68

[Abstract] [PDF Full-Text (820 KB)] **IEEE CNF**

---

2 **SystemC based architecture exploration of a 3D graphic processor**
Kogel, T.; Wieferink, A.; Meyr, H.; Kroll, A.;
Signal Processing Systems, 2001 IEEE Workshop on , 26-28 Sept. 2001
Pages:169 - 176

[Abstract] [PDF Full-Text (472 KB)] **IEEE CNF**

---

3 **Monitoring timing behavior of multi-task programs running on industrial computers**
Hassapis, G.;
Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE , Volume: 3 , 21-23 May 2001
Pages:1485 - 1490 vol.3

[Abstract] [PDF Full-Text (388 KB)] **IEEE CNF**

---

4 **Virtual liver biopsy: image processing and 3D visualization**
Agrafiotis, D.; Jones, M.G.; Nikolov, S.; Halliwell, M.; Bull, D.; Canagarajah, N.;
Image Processing, 2001. Proceedings. 2001 International Conference on , Volume: 2 , 7-10 Oct. 2001
Pages:331 - 334 vol.2

[Abstract] [PDF Full-Text (392 KB)] **IEEE CNF**

Terms used **trace start address real time branch** Found **10,481** of **140,980**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A real-time microprocessor debugging technique

Charles R. Hill

March 1983 **Proceedings of the symposium on High-level debugging**, Volume 18 , 8 Issue 8 , 4

Full text available: pdf(380.29 KB) Additional Information: full citation, abstract, references

This note describes RED, a remotely executed debugger capable of generating a real-time source level trace history of a high level language program executing on a microprocessor. The trace history consists of a display of the source statements of each basic block executed, annotated by the time at which execution of that block began. Basic blocks are traced rather than statements to reduce sampling bandwidth requirements while still retaining the ability to record the essential logical flow of p ...

2 Trace-driven memory simulation: a survey

Richard A. Uhlig, Trevor N. Mudge

June 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 2

Full text available: pdf(636.11 KB) Additional Information: full citation, abstract, references, citings, index terms, review

As the gap between processor and memory speeds continues to widen, methods for evaluating memory system designs before they are implemented in hardware are becoming increasingly important. One such method, trace-driven memory simulation, has been the subject of intense interest among researchers and has, as a result, enjoyed rapid development and substantial improvements during the past decade. This article surveys and analyzes these developments by establishing criteria for evaluating trac ...

Keywords: TLBs, caches, memory management, memory simulation, trace-driven simulation

3 Techniques for efficient inline tracing on a shared-memory multiprocessor

S. J. Eggers, David R. Keppel, Eric J. Koldinger, Henry M. Levy

April 1990 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems**, Volume 18 Issue 1

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms

While much current research concerns multiprocessor design, few traces of parallel programs are available for analyzing the effect of design trade-offs. Existing trace collection methods have serious drawbacks: trap-driven methods often slow down

program execution by more than 1000 times, significantly perturbing program behavior; microcode modification is faster, but the technique is neither general nor portable. This paper describes a new tool, called MPTRACE, for collecting tr ...

4 Synthesis of customized loop caches for core-based embedded systems

Susan Cotterell, Frank Vahid

November 2002 **Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(92.19 KB) Additional Information: full citation, abstract, references, citings, index terms

Embedded system programs tend to spend much time in small loops. Introducing a very small loop cache into the instruction memory hierarchy has thus been shown to substantially reduce instruction fetch energy. However, loop caches come in many sizes and variations -- using the configuration best on the average may actually result in worsened energy for a specific program. We therefore introduce a loop cache exploration tool that analyzes a particular program's profile, rapidly explores the possib ...

Keywords: architecture tuning, customized architectures, embedded systems, estimation, instruction fetching, loop cache, low energy, low power, memory hierarchy, synthesis, tuning

5 No. 1A ESS Laboratory Support System - erasable flag facility

Buyansky, Donald V., Schatz, James W.

September 1982 **Proceedings of the 6th international conference on Software engineering**

Full text available: pdf(623.35 KB) Additional Information: full citation, abstract, references, index terms

A Laboratory Support System (LSS) is provided in all No. 1A Electronic Switching System (ESS) test laboratories to support the ESS software development process. The LSS tools are utilized by the developers to accomplish ESS program database management, lab load administration, lab load generation, and program testing and debugging in the laboratory environment. This document describes one of the LSS testing and debugging tools: the No. 1A ESS Laboratory Erasable Flag System. This facility i ...

6 Address trace compression through loop detection and reduction

E. N. Elnozahy

May 1999 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 27 Issue 1

Full text available: pdf(226.94 KB) Additional Information: full citation, references, index terms

Keywords: address traces, compression, control flow analysis, traces

7 Performance evaluation and improvement of a dynamically microprogrammable computer with low-level parallelism

Shinji Tomita, Kiyoshi Shibayama, Toshiaki Kitamura, Hiroshi Hagiwara

November 1980 **Proceedings of the 13th annual workshop on Microprogramming**

Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, references, citings, index terms

A new microprogrammable computer with low-level parallelism was built and has been utilized as a research vehicle for solving different classes of research-oriented applications such as real-time processings on static/dynamic images, pictures and signals, and emulations of both existing and virtual machines including high

(intermediate) level language machines. The design goal of a research-oriented computer, QA-1, was to achieve a high degree of processing power and system flexi ...

8 Task scheduling and real-time: Extending STI for demanding hard-real-time systems

Benjamin Welch, Shobhit Kanaujia, Adarsh Seetharam, Deepaksrivats Thirumalai, Alexander G. Dean

October 2003 **Proceedings of the international conference on Compilers, architectures and synthesis for embedded systems**

Full text available: pdf(618.51 KB) Additional Information: full citation, abstract, references, index terms

Software thread integration (STI) is a compilation technique which enables the efficient use of an application's fine-grain idle time on generic processors without special hardware support. With STI, a primary function (with real-time requirements on specific instructions) is automatically interleaved with a secondary function to create a single implicitly multithreaded function which minimizes context switching and hence both improves performance and also offers very fine-grain concurrency.In t ...

Keywords: AVR, NTSC video, STIGLitz, embedded systems, fine-grain concurrency, hardware-to-software migration, post-pass compiler, software thread integration

9 Performance estimation of embedded software with instruction cache modeling

Yau-Tsun Steven Li, Sharad Malik, Andrew Wolfe

July 1999 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 4 Issue 3

Full text available: pdf(171.05 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Embedded systems generally interact in some way with the outside world. This may involve measuring sensors and controlling actuators, communicating with other systems, or interacting with users. These functions impose real-time constraints on system design. Verification of these specifications requires computing an upper bound on the worst-case execution time (WCET) of a hardware/software system. Furthermore, it is critical to derive a tight upper bound on WCET in order to make efficient u ...

10 Memory hierarchy: Polynomial-time algorithm for on-chip scratchpad memory partitioning

Federico Angiolini, Luca Benini, Alberto Caprara

October 2003 **Proceedings of the international conference on Compilers, architectures and synthesis for embedded systems**

Full text available: pdf(215.57 KB) Additional Information: full citation, abstract, references, index terms

Focusing on embedded applications, scratchpad memories (SPMs) look like a best-compromise solution when taking into account performance, energy consumption and die area. The main challenge in SPM design is mapping memory locations to scratchpad locations. This paper describes an algorithm to optimally solve such a mapping problem by means of Dynamic Programming applied to a synthesizable hardware architecture. The algorithm works by mapping segments of external memory to physically partitioned b ...

Keywords: design automation, dynamic programming, embedded design, memory hierarchy, partitioning algorithm, power saving, scratchpad memory

11 Application-driven synthesis of core-based systems

Darko Kirovski, Chunho Lee, Miodrag Potkonjak, William Mangione-Smith

November 1997 **Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(176.09 KB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

We developed a new hierarchical modular approach for synthesis of area-minimal core-based data-intensive systems. The optimization approach employs a novel global least-constraining most-constrained heuristic to minimize the instruction cache misses for a given application, instruction cache size and organization. Based on this performance optimization technique, we constructed a strategy to search for a minimal-area processor core, and an instruction and data cache which satisfy the performance ...

Keywords: application-specific system-level synthesis, system modeling and performance evaluation, cache line coloring

12 A Diagnostic Emulator for HEAO software development

Peter H. Beer, Kenneth J. Hupf

August 1976 **Proceedings of the fourth symposium on Simulation of computer systems**

Full text available: pdf(701.56 KB)

Additional Information: full citation, abstract, references, index terms

Diagnostic Emulation is the application of microprogramming to the emulation of an operational computer to support software development and verification for that computer. A conventional technique, Interpretive Computer Simulation (ICS), has been used for many years in support of such software development and verification efforts. The ICS method is becoming less cost effective. For the development of attitude control software for NASA's High Energy Astronomical Observatory (HEAO) diagnostic ...

13 Static grouping of small objects to enhance performance of a paged virtual memory

James W. Stamos

May 1984 **ACM Transactions on Computer Systems (TOCS)**, Volume 2 Issue 2

Full text available: pdf(1.79 MB)

Additional Information: full citation, references, citings, index terms, review

Keywords: Smalltalk, initial placement, object-oriented, paging, programing restructuring, reference trace compression, static grouping, virtual memory

14 The System Simulators - a modular approach to systems modeling

H. F. Hertel, R. A. Merikallio

June 1974 **Proceedings of the 1974 symposium on Simulation of computer systems**

Full text available: pdf(997.71 KB)

Additional Information: full citation, abstract, references, index terms

As computing systems and their applications become progressively more complex, the prediction of the performance of a proposed computer system or the effects of changes to an existing system likewise becomes more difficult. Fortunately there has been steady growth in performance evaluation technology to meet these expanding systems analysis requirements. The most desirable predictive tools continue to be analytical mathematical models which usually provide a comprehensive theoretical unders ...

15 Efficient power co-estimation techniques for system-on-chip design

Marcello Lajolo, Anand Raghunathan, Sujit Dey

January 2000 **Proceedings of the conference on Design, automation and test in Europe**

Full text available: pdf(124.92 KB) Publisher Site Additional Information: full citation, references, citings, index terms

16 Session 10B: Power saving techniques for embedded processors: I-CoPES: fast instruction code placement for embedded systems to improve performance and energy efficiency

Sri Parameswaran, Jörg Henkel

November 2001 **Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(170.07 KB) Additional Information: full citation, abstract, references, index terms

The ratio of cache hits to cache misses in a computer system is, to a large extent, responsible for its characteristics such as energy consumption and performance. In recent years energy efficiency has become one of the dominating design constraints, due to the rapid growth in market share for mobile computing/communication/internet devices.In this paper we present a novel fast constructive technique that relocates the instruction code in such a manner into the main memory that the cache is util ...

17 A systematic approach to advanced debugging: incremental compilation

Peter Fritzson

March 1983 **Proceedings of the symposium on High-level debugging**, Volume 8 , 18 Issue 4 , 8

Full text available: pdf(664.22 KB) Additional Information: full citation, abstract, references

This paper presents two topics: Implementation of a debugger through use of an incremental compiler, and techniques for fine-grained incremental compilation. Both the debugger and the compiler are components of the highly Integrated programming environment DICE (Distributed Incremental Compiling Environment) which alms at providing programmer support in the case where the programming environment resides in a host computer and the program. Is running on a target computer that is connected to the ...

18 On the usefulness of type and liveness accuracy for garbage collection and leak detection

Martin Hirzel, Amer Diwan, Johannes Henkel

November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 6

Full text available: pdf(684.85 KB) Additional Information: full citation, abstract, references, index terms

The effectiveness of garbage collectors and leak detectors in identifying dead objects depends on the *accuracy* of their reachability traversal. Accuracy has two orthogonal dimensions: (i) whether the reachability traversal can distinguish between pointers and nonpointers (*type accuracy*), and (ii) whether the reachability traversal can identify memory locations that will be dereferenced in the future (*liveness accuracy*). This article presents an experimental study of the impo ...

Keywords: Conservative garbage collection, leak detection, liveness accuracy, program analysis, type accuracy

19 Algorithms for a self-tuning microprogrammed computer

K. A. El-Ayat, J. A. Howard

October 1977 **Proceedings of the 10th annual workshop on Microprogramming**

Full text available: pdf(438.57 KB) Additional Information: full citation, abstract, references, index terms

Manual tuning techniques are widely applied but are generally slow, costly and require a great deal of expertise. This paper addresses the problem of automatically tuning the virtual architecture of a microprogrammed computer by microprogramming techniques. Two algorithms are presented to automate the tuning process. The algorithms are implemented on the same dynamic microprogrammed computer that executes the given application. After execution of the program the algorithms are invoked and a ...

20 A multiprocessing system for the direct execution of LISP

Rhon Williams
August 1978 , Volume 10 , 13 , 7 Issue 1 , 2 , 2

Full text available: pdf(691.35 KB) Additional Information: full citation, abstract, references

Current implementations were found to be impractical for airborne use due to LISP's incompatability with conventional computer architectures. Direct execution of LISP with tasks distributed between three processors, seemed to be a workable solution. The language was analyzed, and a special token was devised, using a descriptor with a single pointer. Through careful distribution of responsibilities, control and data flow between the processors was minimized. Significant memory savings resulted fr ...